

Informatica per Odontoiatria



Basi di Dati

2018/19 - Dott. Lorenzo Caruso

Informatica per Odontoiatria

Un **database** (DB) è una collezione di informazioni organizzata in gruppi, che consentono un semplice recupero delle stesse.

2

Nella vita quotidiana esistono innumerevoli esempi di database. Un'agenda telefonica, un vocabolario o un catalogo di viaggi, sono tutti esempi di archivi di dati. Il prelievo da un bancomat o il noleggio in un videodistributore automatico, sono esempi di utilizzo di database (in questo caso gestiti da computer).

L'Informazione nelle Organizzazioni

L'informazione è una componente fondamentale in **qualsiasi** settore, e le tecnologie che ne supportano la gestione da strumento accessorio sono diventate un **bene strategico**

3

Società dell'Informazione, Neteconomy, e-Business

La crescente importanza dell'informazione

- ▶ Un venditore vuole sapere quali clienti non ha visitato nell'ultimo mese e quali prodotti questi clienti hanno già acquistato per programmare le visite della prossima settimana e per sapere quali prodotti può promuovere e quali no.
- ▶ Il responsabile dell'ufficio marketing vuole sapere quali clienti hanno già acquistato il prodotto "X" perché vuole proporre loro l'acquisto di un accessorio di tale prodotto.

4

Il responsabile dell'ufficio del personale vuole sapere se c'è qualcuno nella filiale di Milano con le competenze richieste per coprire una funzione nella nuova filiale di Roma e se gli eventuali candidati hanno espresso la disponibilità a trasferirsi.

Il direttore di una filiale di banca vuole sapere quali clienti sono frequentemente in ritardo nei pagamenti per ottimizzare la procedura di gestione dei prestiti.

Il responsabile di un'indagine statistica sui laureati dell'Università della Calabria vuole conoscere per ciascuna facoltà la percentuale dei laureati con un

voto maggiore di '105'.

Il presidente di un corso di laurea vuole sapere la percentuale degli studenti del 1° anno che possono iscriversi al secondo anno e la media dei loro voti.

I responsabili della Columbia Pictures vogliono conoscere gli incassi dei film interpretati da Tom Cruise per decidere se continuare ad investire sull'attore oppure no.

Archiviare Informazione

... per renderla reperibile:

Efficacemente
(= trovare quello che si cerca e non altro)

Efficientemente
(= a costi contenuti)

Anche da altri
(mediante criteri condivisi)

Come si fa?

...introducendo meta-informazione

es. un codice di classificazione, l'ordine alfabetico, ... la Memoria

Le tre strategie:

archiviazione e reperimento secondo criteri **soggettivi**



archiviazione secondo criteri **condivisi** e reperimento **manuale**



archiviazione secondo criteri **condivisi** e reperimento **automatico**



Un esempio “dal punto di vista dell’utente”

“Mostrami l’elenco dei dipendenti della nostra azienda che hanno frequentato un corso di formazione sul nostro programma gestionale e che abbiano accumulato almeno tre anni di esperienza nell’uso di quel sistema”

Un esempio “dal punto di vista dell’utente”

- ▶ “dovrebbe esserci qualche informazione a riguardo in quello scaffale...”



- ▶ “dovrei trovarli tra le schede del personale sotto la categoria « corsi »”



- ▶ “... eccoli ...”



Cosa significa gestire Informazione?

Il concetto di gestire informazione può tradursi in numerose attività, fra cui:

- ▶ Creare
- ▶ Acquisire
- ▶ Elaborare
- ▶ Archiviare
- ▶ Trasmettere
- ▶ Presentare

Creo l'informazione (Es: Rubrica telefonica – creo il DB),

Acquisisco l'informazione (Inserisco le informazioni nel DB),

Elaboro le Informazioni (ad esempio calcolo l'età delle persone),

Archivio le informazioni (memorizzo e salvo i dati),

Trasmetto le informazioni (possono essere condivise),

Presentare le informazioni (Rendo le informazioni leggibili e presentabili)

Il sistema informativo

Nelle attività umane, le informazioni vengono gestite (registrate e scambiate) in forme diverse:

- ▶ idee informali
- ▶ linguaggio naturale (scritto o parlato, formale o colloquiale, in una lingua o in un'altra)
- ▶ disegni, grafici, schemi
- ▶ numeri e codici

e su vari supporti

- ▶ memoria umana, carta, dispositivi elettronici

Risorse di una organizzazione

Le risorse di una organizzazione:

- persone
- denaro
- materiali
- dati e informazioni (sistema informativo)

Risorse di una organizzazione

Funzioni di un sistema informativo

- raccolta, acquisizione delle informazioni
- archiviazione, conservazione delle informazioni
- elaborazione delle informazioni
- distribuzione, scambio di informazioni
- il concetto di “sistema informativo” è indipendente da qualsiasi forma di automatizzazione

Sistema informatico

porzione automatizzata del sistema informativo.

Il sistema informatico è la parte del sistema informativo che gestisce informazioni per mezzo della tecnologia informatica



Gestione delle Informazioni

Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo forme di organizzazione e codifica delle informazioni.

Ad esempio, nei servizi anagrafici si è iniziato con registrazioni discorsive e poi

- ▶ nome e cognome
- ▶ estremi anagrafici
- ▶ codice fiscale

Dati e Informazioni

Nei sistemi informatici (e non solo), le informazioni vengono rappresentate in modo essenziale: attraverso i **dati**

Dal Vocabolario della lingua italiana:

- ▶ **informazione**: notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere.
- ▶ **dato**: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione; (in informatica) elementi di informazione costituiti da simboli che debbono essere elaborati.

Dati e Informazioni

I dati hanno bisogno di essere interpretati

Esempio

'Mario' '2750' su un foglio di carta sono due dati.

Se il foglio di carta viene fornito in risposta alla domanda "A chi mi devo rivolgere per il problema X; qual è il suo numero interno di telefono? ", allora i dati possono essere interpretati per fornire **informazione** e arricchire la **conoscenza**.


Basi di Dati

- ▶ (accezione generica, **metodologica**)
 - Insieme organizzato di dati utilizzati per il supporto allo svolgimento delle attività di un ente (azienda, ufficio, persona)
- ▶ (accezione specifica, **metodologica e tecnologica**)
 - insieme di dati gestito da un DBMS

DataBase Management System — DBMS

Sistema (o prodotto software) in grado di gestire **collezioni di dati**

Esso facilita l'inserimento, la memorizzazione, la registrazione ed il **reperimento dei dati.**



18

che siano (anche):

grandi (di dimensioni (molto) maggiori della memoria centrale dei sistemi di calcolo utilizzati)

persistenti (con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano)

condivise (utilizzate da applicazioni diverse) garantendo **affidabilità** (resistenza a malfunzionamenti hardware e software)

privatezza (con una disciplina e un controllo degli accessi).

Come ogni prodotto informatico, un DBMS deve essere **efficiente** (utilizzando al meglio le risorse di spazio e tempo del sistema) ed **efficace** (rendendo produttive le attività dei suoi utilizzatori).

Alcuni DBMS in commercio

- ▶ MS Access (Microsoft)
- ▶ DB2 (IBM)
- ▶ Oracle
- ▶ SQLServer

19

DB2 è un Relational Database Management System ([RDBMS](#)) della [IBM](#). La sua prima versione risale al [1983](#); secondo molti, è stato il primo prodotto a utilizzare il linguaggio [SQL](#) ed è stato scritto in [linguaggio C](#) e [C++](#)

Oracle è uno tra i più famosi [database management system](#) (DBMS), cioè sistema di gestione di basi di dati, scritto in [linguaggio C](#).

Oracle fa parte dei cosiddetti [RDBMS](#) (*Relational DataBase Management System*) ovvero di sistemi di database basati sul [Modello relazionale](#) che si è affermato come lo [standard](#) dei [database](#) dell'ultimo decennio.

Microsoft SQL Server è un [DBMS](#) relazionale (*Relational Database Management System* [RDBMS](#)), prodotto da [Microsoft](#). Nelle prime versioni era utilizzato per [basi dati](#) medio-piccole, ma a partire dalla versione 2000 è stato utilizzato anche per la gestione di basi dati di grandi dimensioni.

Condivisione

- ▶ Ogni organizzazione (specie se grande) è divisa in settori o comunque svolge diverse attività
- ▶ A ciascun settore o attività corrisponde un (sotto)sistema informativo
- ▶ Possono esistere sovrapposizioni fra i dati di interesse dei vari settori
- ▶ Una base di dati è una risorsa **integrata**, condivisa fra i vari settori

Possibili problemi

- ▶ **Ridondanza:** informazioni ripetute
- ▶ **Rischio di incoerenza:** le versioni possono non coincidere

Le basi di dati sono condivise

Una base di dati è una risorsa **integrata, condivisa** fra le varie applicazioni

▶ **conseguenze**

- Attività diverse su dati in parte condivisi: meccanismi di **autorizzazione**
- Attività multi-utente su dati condivisi: controllo della **concorrenza**

Descrizioni dei dati nei DBMS

Descrizioni e rappresentazioni dei dati a livelli diversi

- ▶ permettono l'**indipendenza dei dati** dalla rappresentazione fisica: i programmi fanno riferimento alla struttura a livello più alto, e le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi

Precisiamo attraverso il concetto di **modello dei dati**

Modello dei dati

- ▶ insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- ▶ componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- ▶ come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ▶ ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

24

Nell'ambito delle [basi di dati](#), il **modello dei dati** rappresenta un insieme di strumenti concettuali, detto "*formalismo*", che consta di tre componenti essenziali:

Insieme di strutture dati, con operatori opportuni.

Notazione per specificare i dati tramite le strutture dati del modello.

Insieme di operazioni per manipolare i dati.

Qualsiasi modello dei dati deve risolvere due principali quesiti: Come rappresentare le [entità](#) e i loro attributi.

Come rappresentare le associazioni.

Nel primo caso la maggioranza dei modelli usa strutture come i [record](#), in cui ogni componente rappresenta un attributo.

Nel secondo caso i modelli differiscono notevolmente nella ricerca di una rappresentazione, pertanto si possono proporre diversi esempi come le strutture, i valori, i [puntatori](#), ecc.

Schemi e istanze

In ogni base di dati esistono:

- ▶ lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura (*le intestazioni delle tabelle*)
- ▶ l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente (*il "corpo" di ciascuna tabella*)

Due tipi di modelli

- ▶ **modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione ... il più noto è il modello Entity-Relationship (ER)
- ▶ **modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati
 - utilizzati dai programmi
 - indipendenti dalle strutture fisiche
 - esempi: relazionale, reticolare, gerarchico, a oggetti

26

Le fasi della progettazione si distinguono per il grado di astrattezza che le caratterizzano.

Si passa dalla definizione del modello concettuale (scelta del DBMS e, se si trattasse di dati relazionali, definizione dello schema Entità-Relazione E.R.), alla fase di definizione del modello logico (definizione di attributi e campi di collegamento) per giungere infine alla realizzazione fisica del DataBase con il modello Fisico.

Il modello concettuale

Fornisce una rappresentazione semplificata della realtà. Il grado di semplificazione dipende dalle scelte che si impone chi sviluppa il modello.

Si tratta quindi di definire un insieme di dati presenti in natura e che rappresentano la natura stessa delle informazioni che si vogliono archiviare.

I dati poi devono essere individuati in modo tale che rispondano allo scopo per il quale si crea il modello. Non esistono pertanto delle regole prefissate per

l'individuazione dei dati e per la loro selezione: ad uno scopo diverso corrisponde un insieme di dati diverso e quindi un modello concettuale diverso. A seconda quindi delle scelte che si operano si otterranno modelli concettuali diversi

Modello logico

Questo modello permette di definire l'organizzazione dei dati, le modalità di accesso e le viste, cioè gli schemi esterni.

In questa fase si svilupperà il modello logico relativo al tipo di database che si ritiene più adatto alle informazioni da archiviare. Il modello logico dipende quindi dal tipo di DBMS che i dati richiedono. Nel caso di un DataBase di tipo relazionale, il modello logico si occuperà di stabilire:

1. gli **attributi** relativi alle entità e
2. le **chiavi** interne (chiavi primarie) ed esterne per la concretizzazione delle relazioni emerse dal diagramma E-R

Modello Concettuale – Perché

Con un modello concettuale è possibile concentrarsi inizialmente sui soli aspetti importanti

- ▶ senza bisogno di specificare come gli oggetti del DB devono essere tra loro in relazione
- ▶ ma cosa deve essere posto in relazione

Ogni modello concettuale prevede efficaci rappresentazioni grafiche, utili anche per documentazione e comunicazione

Modello Entità Relazioni

- ▶ Ci permette di descrivere i dati coinvolti in una organizzazione del mondo reale in termini di oggetti e delle loro relazioni
- ▶ Esso fornisce concetti che ci permettono di passare da una descrizione informale di ciò che gli utenti chiedono alla loro BD ad una descrizione precisa e più dettagliata che può essere implementata in un DBMS
- ▶ Ha una rappresentazione grafica (Diagrammi E/R)

Il Modello Entità-Relazione

Si deve a Peter Chen

- ▶ Nel 1976 propone la tecnica Entity-Relationship
- ▶ Insieme alla tecnica, propose una notazione grafica detta diagramma E/R (ERD)
- ▶ Oggi il diagramma E/R è il più utilizzato per immediatezza e semplicità

Il Modello Entità-Relazione

Scopo del modello Entità-Relazione (E-R):

- ▶ permettere la descrizione dello schema concettuale di una situazione reale senza preoccuparsi dell'efficienza o della progettazione del database fisico.

Lo schema Entità-Relazione così costruito sarà poi tradotto in uno schema logico di un modello logico dei dati, ad esempio quello relazionale, che al momento è il più diffuso.

Il Modello Entità-Relazione

Costrutti fondamentali:

- ▶ Entità (Entity)
- ▶ Relazione (Relationship)
- ▶ Attributi
 - di Entità
 - di Relazione

Il Modello Entità-Relazione

Il modello Entità - Relazioni schematizza la realtà in termini di Entità e di Relazioni fra di esse.

▶ Entità

- è un oggetto del mondo reale che si distingue da altri oggetti
- è difficile da definire a parole, ma operativamente si comprende
- Es. Enrico Rossi con un dato CF è una entità, poiché identifica una particolare persona
- può essere concreto, come un libro
- può essere astratta, come una vacanza

Modello Entità-Relazione

Entity-Set:

- ▶ è un insieme di entità dello stesso tipo, identificate da un nome
 - Es. l'insieme delle persone che hanno un conto in banca può essere definito come l'Entity-Set Cliente
- ▶ gli Entity-Set possono non essere disgiunti
 - definendo gli Entity-Set Cliente ed Impiegato, posso avere un impiegato che è anche cliente della banca (ossia ha un conto in banca)



Modello Entità-Relazione

Attributo:

- è una proprietà dell' entità, di interesse ai fini dell'applicazione
- una entità è identificata da un insieme di valori detti attributi
 - per l'entità Cliente, attributi possono essere *il nome del cliente, il codice fiscale, l'indirizzo*
 - per l'entità Conto, possono essere *il numero di conto ed il saldo*
 - per l'entità Persona, *nome, cognome, codice fiscale, ...*
- Tutte le entità di un dato insieme di entità hanno gli stessi attributi
- Si ricavano dal documento di analisi dei requisiti

Modello Entità-Relazione

Il Dominio di un attributo rappresenta l'insieme dei valori assunti dall'attributo

- ▶ per l'attributo nome del cliente, il Dominio può essere l'insieme delle stringhe di lunghezza 20 caratteri
- ▶ per il numero di conto, può essere l'insieme di tutti i numeri interi positivi •
- ▶ (tipicamente, interi, caratteri, stringhe, ecc.)

Formalmente, un attributo è una funzione definita da un EntitySet in un dominio

Modello Entità-Relazione

Una Entità è descritta dall'insieme delle coppie (attributo, valore)

- ▶ una particolare entità cliente è descritta dall'insieme {(nome, Rossi), (C.F., 12345656SE31), (Indirizzo, Via Della Vigna Nuova Firenze) }
- ▶ tale entità rappresenta il cliente Rossi con CF 1234566S31 che risiede a Firenze in Via Della Vigna Nuova
- ▶ L'Entity Set corrisponde alla nozione di tipo di dato astratto in un linguaggio di programmazione
- ▶ L'entità corrisponde al concetto di variabile

Modello Entità-Relazione

Relazione

- ▶ una relazione è una associazione fra 2 o piu' entità:
 - data l'entità Rossi e l'entità conto corrente numero 1278 (identificata da numero di conto corrente e saldo), si può definire una relazione che associa le due entità
 - in tal modo si esprime il fatto che Rossi è un cliente con numero di conto 1278
- ▶ una relazione viene identificata con un nome

Modello Entità-Relazione

Relationship – Set (Insieme di Relazioni)

- ▶ insieme di relazioni dello stesso tipo
 - se ne può dare una formalizzazione

- ▶ Insieme di Relazioni
 - “Fotografia” dell’insieme di Relazioni in un certo istante

Modello Entità-Relazione

Classificazione degli attributi:

- ▶ **Semplici:** se possono assumere un solo valore
 - es. il nome di un cliente
- ▶ **Multipli:** se prevedono la possibilità di più valori
 - es. l'attributo esami sostenuti per uno studente
- ▶ **Composti:** quando è possibile scomporli in più elementi
 - es. l'indirizzo, che può essere scomposto in via, n. civico, CAP, città
- ▶ **Opzionali:**
 - es. n. patente

Diagrammi Entità-Relazioni

Lo schema concettuale realizzato tramite il modello E-R si rappresenta per mezzo dei diagrammi Entità - Relazioni

- ▶ non esistono notazioni standardizzate per la realizzazione di tali diagrammi, noi ne vedremo una
- ▶ esistono degli strumenti che, adottando una particolare notazione, consentono la realizzazione di tali diagrammi (e molto di più)

Diagrammi Entità-Relazioni

Entità

- ▶ vengono rappresentate tramite rettangoli
 - in realtà viene rappresentato l' Entity Set, non la singola istanza



Attributi

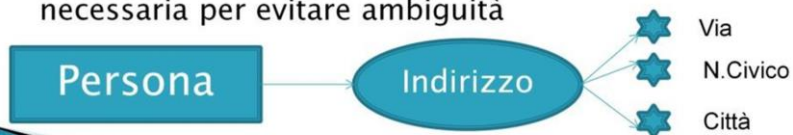
- vengono rappresentati tramite ovali



Modello Entità-Relazione

Attributi composti

- ▶ Sono attributi che si ottengono aggregando altri (sotto) attributi, i quali presentano una forte affinità nel loro uso e significato
 - Es.: via, n. civico, città e CAP formano l'attributo composto indirizzo
 - Nel caso di presenza di più attributi multivalore, la creazione di un attributo composto può rendersi necessaria per evitare ambiguità



Modello Entità-Relazione

Attributi Multipli

- ▶ prevedono la possibilità di più valori
 - es. l'attributo `VotiEsami` sostenuti da uno studente
 - es. l'attributo `TitoliDiStudio` di un impiegato

Due tipi di modelli

- ▶ **modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione ... il più noto è il modello Entity-Relationship (ER)
- ▶ **modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati
 - utilizzati dai programmi
 - indipendenti dalle strutture fisiche
 - esempi: relazionale, reticolare, gerarchico, a oggetti

44

Le fasi della progettazione si distinguono per il grado di astrattezza che le caratterizzano.

Si passa dalla definizione del modello concettuale (scelta del DBMS e, se si trattasse di dati relazionali, definizione dello schema Entità-Relazione E.R.), alla fase di definizione del modello logico (definizione di attributi e campi di collegamento) per giungere infine alla realizzazione fisica del DataBase con il modello Fisico.

Il modello concettuale

Fornisce una rappresentazione semplificata della realtà. Il grado di semplificazione dipende dalle scelte che si impone chi sviluppa il modello.

Si tratta quindi di definire un insieme di dati presenti in natura e che rappresentano la natura stessa delle informazioni che si vogliono archiviare.

I dati poi devono essere individuati in modo tale che rispondano allo scopo per il quale si crea il modello.

Non esistono pertanto delle regole prefissate per l'individuazione dei dati e per la loro selezione: ad uno scopo diverso corrisponde un insieme di dati diverso e quindi un modello concettuale diverso. A seconda quindi delle scelte che si operano si otterranno modelli concettuali diversi

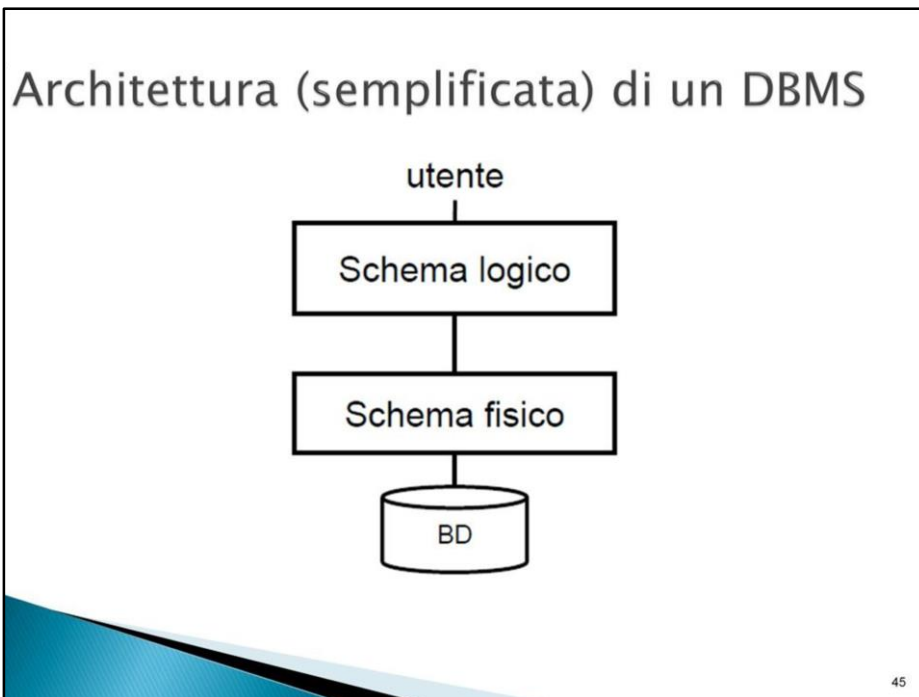
Modello logico

Questo modello permette di definire l'organizzazione dei dati, le modalità di accesso e le viste, cioè gli schemi esterni.

In questa fase si svilupperà il modello logico relativo al tipo di database che si ritiene più adatto alle informazioni da archiviare. Il modello logico dipende quindi dal tipo di DBMS che i dati richiedono.

Nel caso di un DataBase di tipo relazionale, il modello logico si occuperà di stabilire:

1. gli **attributi** relativi alle entità e
2. le **chiavi** interne (chiavi primarie) ed esterne per la concretizzazione delle relazioni emerse dal diagramma E-R



schema logico: descrizione della base di dati nel modello logico (ad esempio, la struttura della tabella)

schema fisico: rappresentazione dello schema logico per mezzo di strutture memorizzazione (file)

Indipendenza dei dati

Il livello logico è indipendente da quello fisico:

- una tabella è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica (che può anche cambiare nel tempo)

Indipendenza dei dati

- ▶ conseguenza della articolazione in livelli
- ▶ l'accesso avviene solo tramite il livello esterno (che può coincidere con il livello logico)
- ▶ due forme:
 - **indipendenza fisica**
 - **indipendenza logica**

Indipendenza fisica

Il livello logico e quello esterno sono indipendenti da quello fisico

- una relazione è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica
- la realizzazione fisica può cambiare senza che debbano essere modificati i programmi

Indipendenza logica

- ▶ il livello esterno è indipendente da quello logico
- ▶ aggiunte o modifiche alle viste non richiedono modifiche al livello logico
- ▶ modifiche allo schema logico che lascino inalterato lo schema esterno sono trasparenti

Linguaggi per basi di dati

Un altro contributo all'efficacia: disponibilità di vari linguaggi e interfacce diverse

50

- linguaggi testuali interattivi (**SQL**)
- comandi (come quelli del linguaggio interattivo) immersi in un linguaggio **ospite** (Pascal, C, Cobol, etc.)
- comandi (come quelli del linguaggio interattivo) immersi in un linguaggio ad hoc, con anche altre funzionalità (p.es. per grafici o stampe strutturate), anche con l'ausilio di strumenti di sviluppo (p. es. per la gestione di maschere)
- con interfacce amichevoli (senza linguaggio testuale, QBE – Query By Example)

SQL, un linguaggio interattivo

QUERY: Quali sono i corsi che si tengono a piano terra?

```
SELECT Corso, Aula, Piano
FROM Aule, Corsi
WHERE Nome = Aula
AND Piano="Terra"
```

Corso	Aula	Piano
Sociologia	SSP2	Terra
Sis. Inf	SSP1	Terra

51

Una volta strutturata e popolata una base di dati, come si estraggono le informazioni?

I DBMS forniscono strumenti (linguaggi) per comporre al DB le richieste (query), anche complesse, di aggregazione

Criteri di selezione e visualizzazione di dati, provenienti anche da tabelle diverse

Interazione non testuale (in MS Access)

Campo:	Espr1: Anagrafica_	Espr2: Anagrafica_Art	Espr3: Anagraf
Tabella:			
Ordinamento:			
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteri:			
Oppure:			

Progettazione e implementazione

Vi sono differenze sostanziali tra Access e gli altri programmi contenuti nel pacchetto MS Office (Word, Excel, PowerPoint).

1. Lavorare con Access prevede necessariamente una parte iniziale di progettazione del database
2. Access non è un software WYSIWYG (What you see is what you get)
3. Esistono due livelli di utilizzo del software Access: •
 1. Colui che progetta e sviluppa la struttura del database, ne implementa tutti gli oggetti e ne controlla il funzionamento
 2. Colui che utilizza il database, inserendo i dati, eseguendo delle ricerche, stampando rapporti,...

Progettazione e implementazione

Le due fasi della creazione di un database sono progettazione e implementazione:

1. Inizialmente analizzeremo i concetti e le regole che si devono seguire per progettare un database
 1. Raccolta e analisi dei requisiti necessari
 2. Progettazione concettuale
 3. Progettazione logica
2. Dopo aver familiarizzato con i concetti di base della progettazione, li utilizzeremo per implementare un database in Microsoft Access

La progettazione deve essere studiata tenendo conto del fatto che Access è un DBMS che si basa sul modello relazionale, permette cioè di mettere in relazione dati che si trovano in tabelle distinte.

Organizzazione dati

La struttura di un DBMS relazionale (come Ms Access) è composta da Tabelle: in esse vengono memorizzate le informazioni del database.

N.B. un DB creato con Access è, solitamente, un unico file con estensione .mdbx, il quale contiene generalmente più tabelle

Ogni tabella ha una struttura che consente la raccolta, l'organizzazione, la memorizzazione e la ricerca di dati

Access: tabelle

- ▶ Una tabella è composta da campi (le colonne) e record (le righe).
- ▶ I campi sono categorie di informazioni come, ad esempio, Nome, Cognome e Data di Nascita.
- ▶ L'insieme dei campi che contiene i dati di una singola voce è chiamato record (istanza).

Tabella: esempio

students

Matricola	Cognome	Nome	Data di Nascita
6554	Rossi	Mario	5/12/1978
8765	Neri	Paolo	3/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	1/2/1978

Record

Campi

Tabella

Tabella: regole

1. Ridurre le Tabelle in Prima Forma Normale: è necessario ricondurre ogni campo della Tabella ad essere "atomico" (non ulteriormente decomponibile)
2. Non creare campi duplicati (cioè campi che hanno nome diverso, ma contengono la stessa informazione)
3. Non creare un campo che ha lo stesso valore in tutti i record, se siete sicuri che tale valore resterà inalterato (es: se nella Tabella Clienti sapessi che i clienti sono tutti di Ravenna sarebbe inutile inserire il campo Città)
4. Ogni record deve essere unico (per essere sicuri che ogni record sia unico, è necessario creare un campo che sia chiave primaria della Tabella, vedi sotto)

Attenzione alla possibilità di valori nulli di diversi tipi!

Chiave primaria

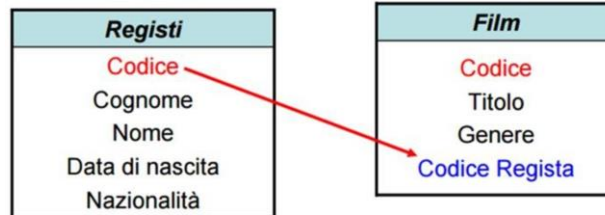
La quarta regola enuncia che una tabella è ben strutturata quando ogni suo record può essere identificato in maniera univoca, ossia quando esiste almeno un campo che non assume mai valori uguali nella tabella.

Il campo utilizzato per identificare univocamente un record si chiama **Chiave Primaria**. Un esempio tipico di chiave primaria è un contatore (cioè un numero che corrisponde a uno e un solo record, e che cresce progressivamente all'inserimento di ogni record) .

N.B.: la chiave primaria di un record non deve mai avere valore nullo (**Primo vincolo di integrità referenziale**)

Chiave importata

Per mettere in relazione due tabelle tra loro, è necessario che abbiano un campo in comune, o meglio, che entrambe abbiano un campo che contiene gli stessi dati. Se in una delle due tabelle questo campo è chiave primaria, allora si dice che quel campo è chiave importata nella seconda tabella.



N.B.: i valori contenuti nella chiave importata devono essere per forza presenti anche tra i valori della chiave primaria! (Secondo vincolo di integrità).

DB: esempio di progettazione

D'ora in poi seguirà una parte di progettazione basata su un esempio reale: un db in grado di conservare informazioni riguardanti film di una collezione privata.

DB film: informazioni da archiviare

- ▶ Il primo passo della progettazione è l'analisi dei requisiti necessari.
- ▶ Poi è necessario decidere quali informazioni archiviare.
- ▶ Supponiamo nel nostro esempio:
Film Registri Attori
- ▶ Il nostro archivio conterrà inizialmente 3 tabelle.

Db film: le tabelle

<i>Film</i>
Titolo
Genere
Anno
Durata
Oscar vinti

<i>Registi</i>
Cognome
Nome
Data di nascita
Nazionalità
Oscar vinti
Nota biografica

<i>Attori</i>
Cognome
Nome
Data di nascita
Nazionalità
Oscar vinti

Db film: le chiavi primarie

<i>Film</i>	<i>Registi</i>	<i>Attori</i>
Codice Titolo Genere Anno Durata Oscar vinti	IDRegista Cognome Nome Data di nascita Nazionalità Oscar vinti Nota biografica	Codice Cognome Nome Data di nascita Nazionalità Oscar vinti

64

Si possono creare anche chiavi primarie multiple, ad esempio nella tabella FILM si potrebbe creare una chiave utilizzando insieme i campi TITOLO-ANNO-DURATA (è estremamente improbabile che siano contemporaneamente uguali in due record). Questa scelta avrebbe il vantaggio di non avere il campo Codice, quindi avrebbe meno informazioni “inutili”, ma a livello di implementazione creerebbe sicuramente più problemi in fase di costruzione del DB con Access.

Le Relazioni

Le tabelle possono essere messe in relazione tra loro, una tabella può essere in relazione con più tabelle. Difficilmente in un Database relazionale avremo tabelle che non sono in relazione con altre.

E' importante, in fase di progettazione, considerare e evidenziare la cardinalità delle relazioni, cioè, per ogni relazione esistente considerare se un record di una tabella è collegato a un solo record di un'altra tabella oppure se è collegato a più record, e le cardinalità possibili sono 3:

Uno a Uno

Uno a Molti

Molti a Molti

Relazioni uno-a-uno

Nel nostro esempio non compaiono relazioni di tipo uno-a-uno.

Alcuni esempi possono essere:

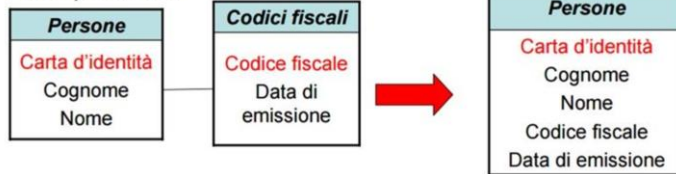
- ▶ Tra persone e codice fiscale
- ▶ Tra automobili e targhe

66

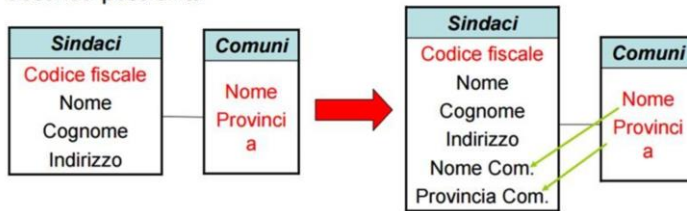
Con una relazione uno-a-uno si tende a inserire i dati in una tabella unica per non creare duplicazione. In certi casi, quando si hanno dati invariati nel tempo e che si vogliono sempre mantenere anche se non associati, si possono creare due tabelle, con una chiave importata in una delle due.

Relazioni uno-a-uno

Prima possibilità



Seconda possibilità

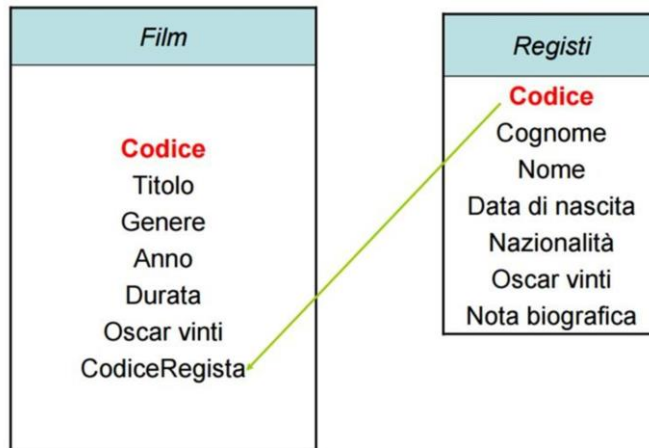


Relazioni uno-a-molti

Nel nostro esempio: a ciascun regista possono essere associati più film, mentre a ciascun film è associato un solo regista.

Per questo tipo di relazioni è necessario creare due tabelle in una delle quali importiamo una chiave dell'altra tabella (N.B.: in particolare importiamo la chiave della tabella con la cardinalità "molti" a quella con la cardinalità "uno")

Relazioni uno a molti



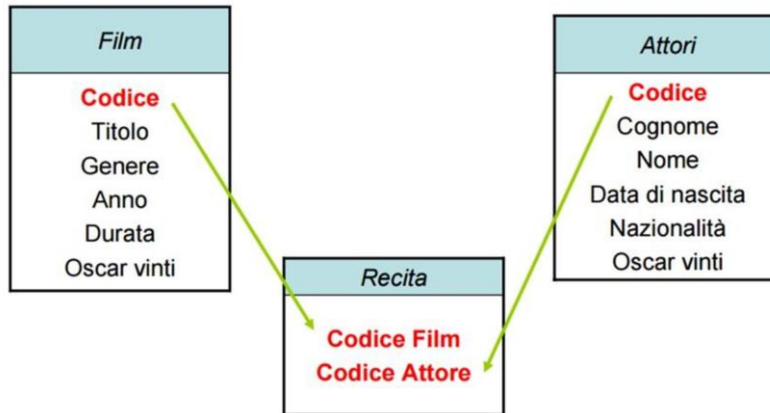
Relazioni molti-a-molti

Nel nostro esempio: a ciascun attore possono essere associati più film, e a ciascun film sono associati più attori.

Per questo tipo di relazioni è necessario creare 3 tabelle: 2 che contengano gli attributi delle due entità e una che contenga entrambe le chiavi primarie delle altre due entità più, eventualmente, altri attributi che non compaiono nelle due tabelle originali.

Nella terza tabella la chiave primaria sarà formata, al minimo, dalle chiavi importate dalle altre tabelle.

Relazioni multi-a-molti



Progettazione DB: Riepilogo

I vari passaggi della progettazione di un DB (secondo il modello relazionale) sono:

- ▶ Raccolta e analisi dei requisiti necessari
- ▶ Decidere quali tabelle creare e quali campi inserire in ogni tabella
- ▶ Decidere quali sono le chiavi primarie nelle tabelle
- ▶ Decidere quali relazioni esistono tra le tabelle
- ▶ Sulla base della tipologia di relazioni valutare la necessità di aggiungere chiavi importate o altre tabelle di collegamento

Implementazione DB

A questo punto l'implementazione tramite Access diventa una serie di operazioni meccaniche di trasposizione

1. Creazione delle **Tabelle**, definendo i **Campi**, **Chiavi primarie** e chiavi importate.
2. Creazione delle **Relazioni**, definendo quali sono i campi che servono ad unire due tabelle
3. Disegnare **Maschere** di inserimento consultazione e gestione per facilitare l'utente ad utilizzare il DB
4. Decidere in quali modi il DB può essere interrogato (tramite le **Query**)
5. Disegnare i possibili Rapporti di dati, per poter stampare o semplicemente consultare i dati ordinati e catalogati con una struttura stabilita (tramite i **Report**)

Esempio di un DB Access

